

A black and white illustration of the RMS Titanic sinking. The ship is tilted at a steep angle, with its bow high in the air and its stern submerged. Large plumes of smoke or steam are rising from the ship's funnels. In the foreground, several lifeboats are visible, some with people inside, and others are empty. The sea is choppy, and the sky is overcast. The overall scene is dramatic and somber.

Pleiten, Pech und PatternTesting

Ein Drama in 5,5 Akten

oliver.boehm@agentes.de
christian.heise@agentes.de



**neugierig?
ängstlich?**



Pleiten

Pech

Pattern
Testing

Pattern
Testing



A red velvet curtain with a gold fringe hangs at the top of the frame, framing a white background.

Prolog

```
...  
declare  
    vertical_veloc_sensor: float;  
    horizontal_veloc_sensor: float;  
    vertical_veloc_bias: integer;  
    horizontal_veloc_bias: integer;  
    ...  
begin  
    declare  
        pragma suppress(numeric_error, horizontal_veloc_bias);  
    begin  
        sensor_get(vertical_veloc_sensor);  
        sensor_get(horizontal_veloc_sensor);  
        vertical_veloc_bias := integer(vertical_veloc_sensor);  
        horizontal_veloc_bias := integer(horizontal_veloc_sensor);  
        ...  
    exception  
        when numeric_error => calculate_vertical_veloc();  
        when others => use_irs1();  
    end;  
end irs2;
```



Laurent DEJAULT

<http://www.flickr.com/photos/lolodoc/3845299578/>



Pattern



Front End

Middle Tier

Back End

<http://www.flickr.com/photos/ann/1329361/>

AntiPattern



Lava-Flow

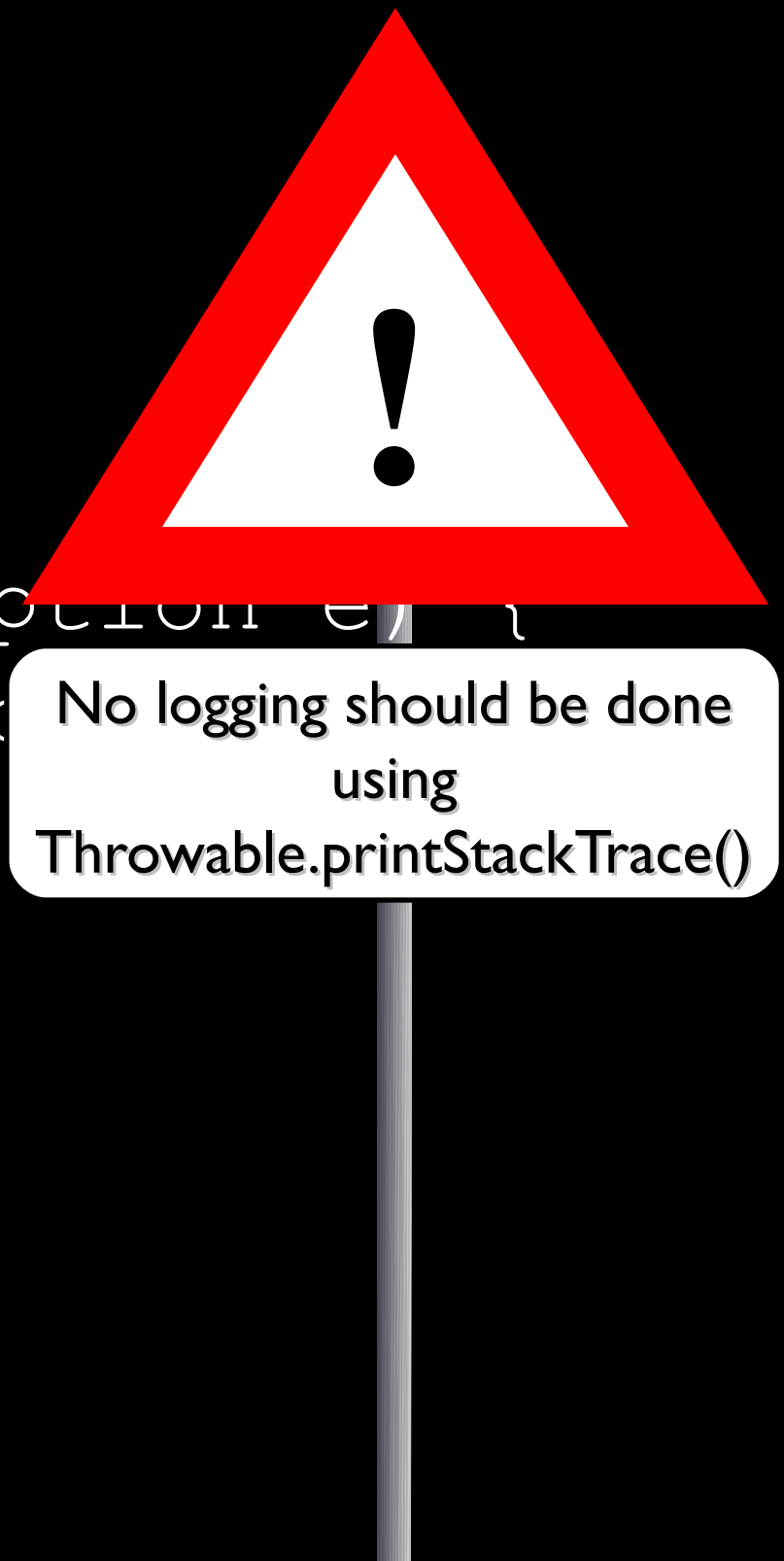


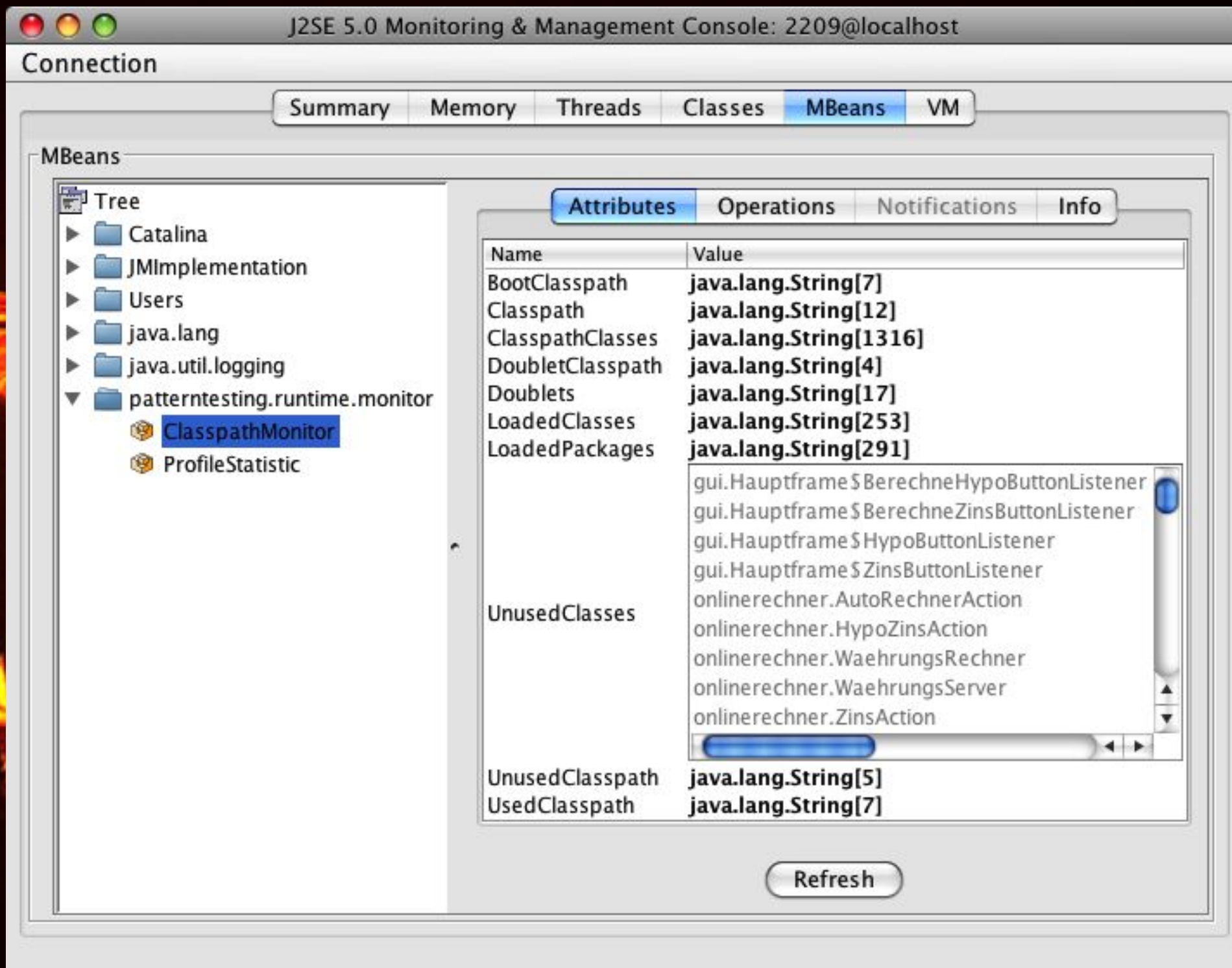
Is. Dan Harkness
WildPhotons.com

```
try {  
    ...  
} catch (MalformedURLException e) {  
    // TODO Auto-generated catch block  
    e.printStackTrace();  
}
```

AntiPattern Testing

```
try {  
    ...  
} catch (MalformedURLException e) {  
    // TODO Auto-generated  
    e.printStackTrace();  
}
```





ProfileStatistic

Label	Unit	LastValue	Hits	Avg	Total
patterntesting.sample.World.main(String[])	ms.	20	1	20	20
patterntesting.sample.World.printHelloAgain()	ms.	0	1	0	0
new patterntesting.sample.World()	ms.	0	0	0	0
patterntesting.sample.World.printBye()	ms.	0	0	0	0



It is not allowed to use the JDBC API from here!

Apache Tomcat/6.0.20 - Error report

⏪ ⏩ +

http://localhost:6060/MafiosoBankPortal/LoginServlet

↻

🔍 Google

Apache Tomcat/6.0.20 - Error report

HTTP Status 500 -

type

Exception report

message

description

The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
java.lang.NullPointerException
    patterntesting.sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:34)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:637)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
```

note

The full stack trace of the root cause is available in the Apache Tomcat/6.0.20 logs.

Apache Tomcat/6.0.20

Login

User

Guiseppe

Password

Login

```
protected void doPost(HttpServletRequest request,
    HttpServletResponse response) throws ServletException, IOException {
    String password = request.getParameter("password");
    String userName = request.getParameter("user");

    User user = LoginService.login(userName, password);
    ServletContext context = getServletContext();
    RequestDispatcher dispatcher = context.getRequestDispatcher("/overview.jsp");
    request.setAttribute("user_login", user.getName());
    dispatcher.include(request, response);
}
```

java.lang.NullPointerException

at sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:35)

at sample.jfs2010.web.LoginServletTest.testDoPost(LoginServletTest.java:52)

A red velvet curtain with a scalloped edge hangs across the top of the frame. The background is a plain, light-colored surface.

Erster Akt: Ein Königreich für einen Test

```
public void testDoPost() throws IOException {  
    LoginServlet servlet = new LoginServlet();  
    MockHttpServletRequest req = new MockHttpServletRequest();  
    MockHttpServletResponse resp = new MockHttpServletResponse();  
    req.setParameter("user", "guiseppa");  
    req.setParameter("password", "topsecret");  
    servlet.doPost(req, resp);  
    log.info("Yeah, no exception!");  
}
```

java.lang.NullPointerException

at sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:35)

at sample.jfs2010.web.LoginServletTest.testDoPost(LoginServletTest.java:52)

A red curtain with a scalloped edge hangs across the top of the image, framing a white background.

Akt 2.0: PatternTesting betritt die Bühne

ohne PatternTesting



service()

doPost()

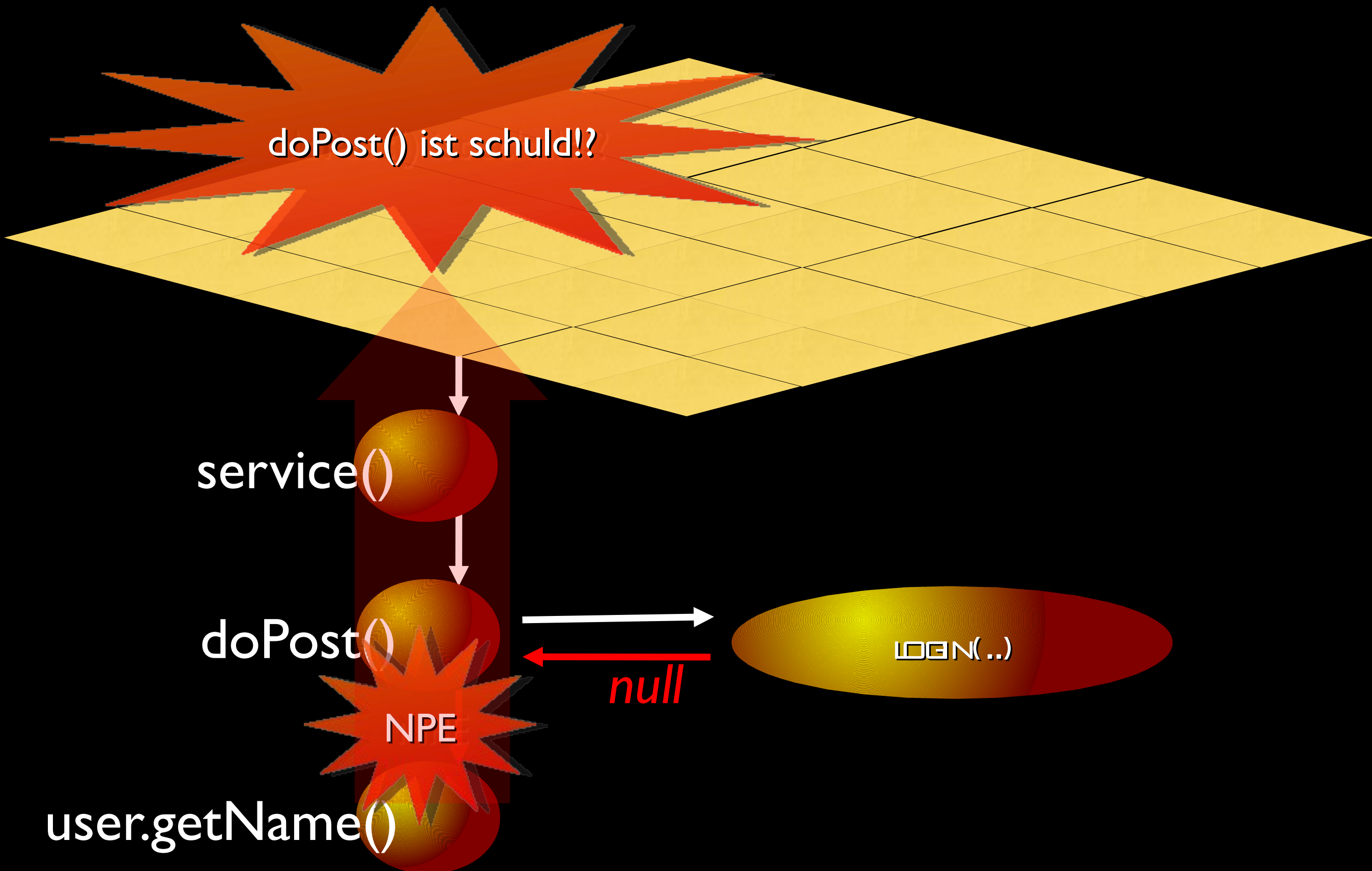
user.getName()

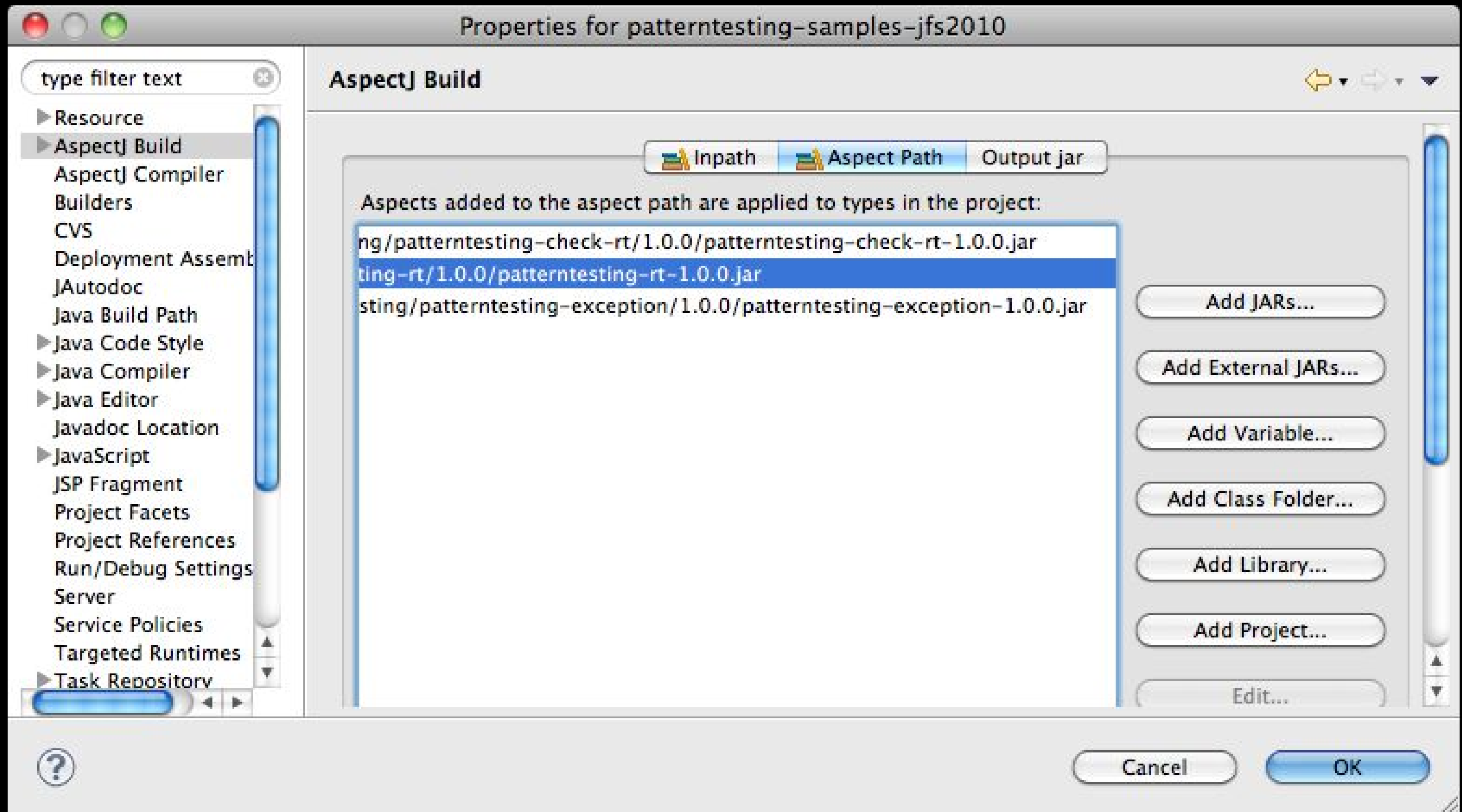


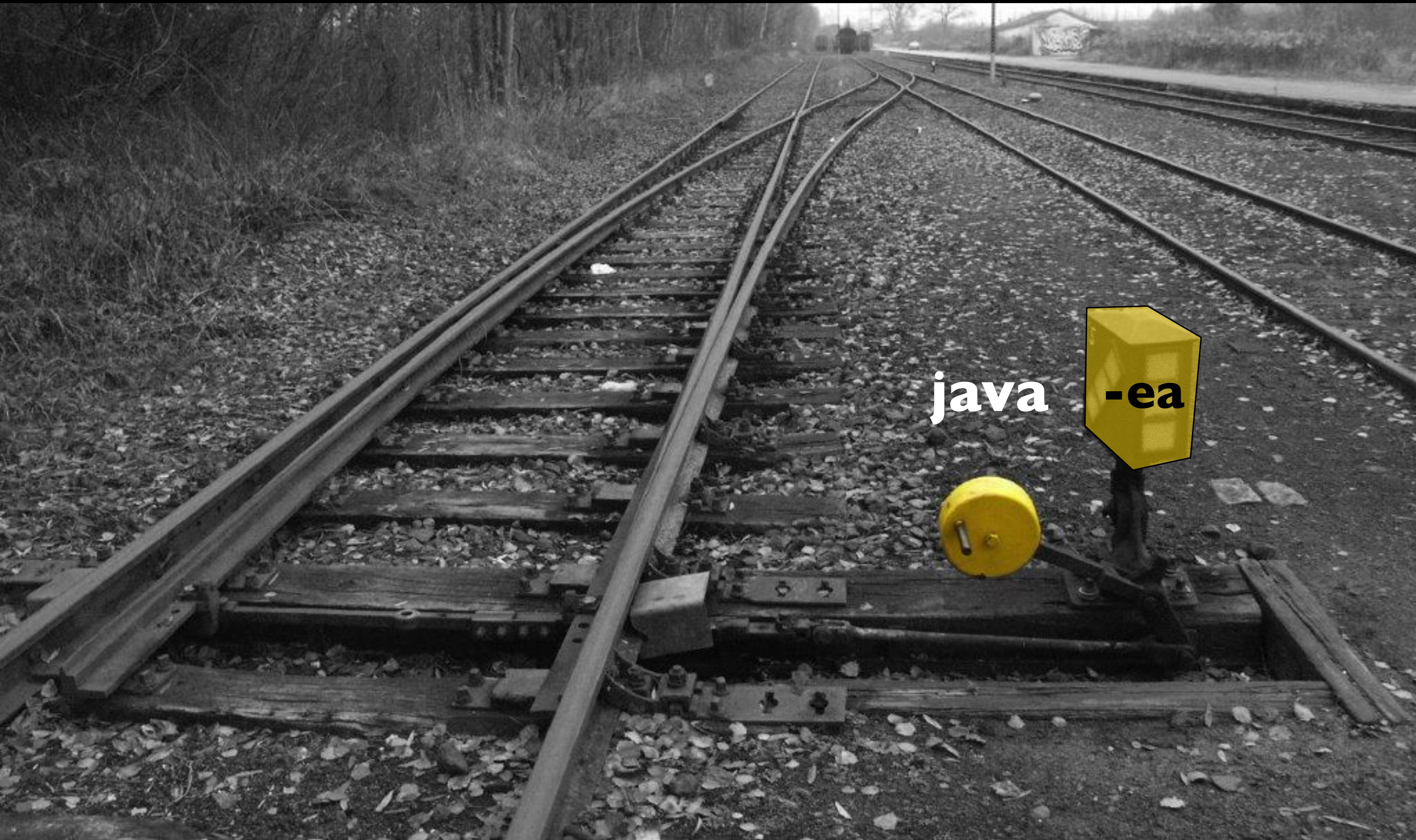
null

LOGN(..)

ohne PatternTesting







java

-ea

mit PatternTesting



service()

doPost()

AssertionError:
login() returns null!

java.lang.AssertionError: **User LoginService.login(String, String) returns null!**

...

at sample.jfs2010.LoginService.login(LoginService.java:62)

at sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:26)

...

```
public static User login(final String name, final String passwd) throws IOException,
    LoginException {
    User user = null;
    URL url = getLoginURL(name);
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    int rc = connection.getResponseCode();
    log.debug("RC=" + rc);
    if (rc != 404) {
        user = new User(name);
    }
    return user;
}
```

```
public static User login(final String name, final String passwd) throws IOException,
    LoginException {
    URL url = getLoginURL(name);
    HttpURLConnection connection = (HttpURLConnection) url.openConnection();
    int rc = connection.getResponseCode();
    log.debug("RC=" + rc);
    if (rc == 404) {
        throw new LoginException(name + ": login failed");
    }
    return new User(name);
}
```



HTTP Status 500 -

type Exception report

message

description The server encountered an internal error () that prevented it from fulfilling this request.

exception

```
javax.servlet.ServletException: javax.security.auth.login.LoginException: guiseppe: login failed
    patterntesting.sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:34)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:637)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
```

root cause

```
javax.security.auth.login.LoginException: guiseppe: login failed
    patterntesting.sample.jfs2010.LoginService.login(LoginService.java:65)
    patterntesting.sample.jfs2010.web.LoginServlet.doPost(LoginServlet.java:28)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:637)
    javax.servlet.http.HttpServlet.service(HttpServlet.java:717)
```

note The full stack trace of the root cause is available in the Apache Tomcat/6.0.20 logs.

A red curtain with a scalloped edge hangs across the top of the frame, framing a white background.

**Pause:
Ein Blick
hinter die Bühne**

Pattern Testing - Bibliothek

LoginService.java

```
class LoginService {  
    login(...){  
        ...  
    }  
    getLoginURL(...){  
        ...  
    }  
}
```

WEAVING

Wunder
Wunder
Wunder

NullPointerException

alle Methoden
am Ende

assert
einfügen

pattern-testing-check-rt

pattern-testing-rt

login
...

• class

assert //

getLoginURL
...

assert //

AOP-Begriffe

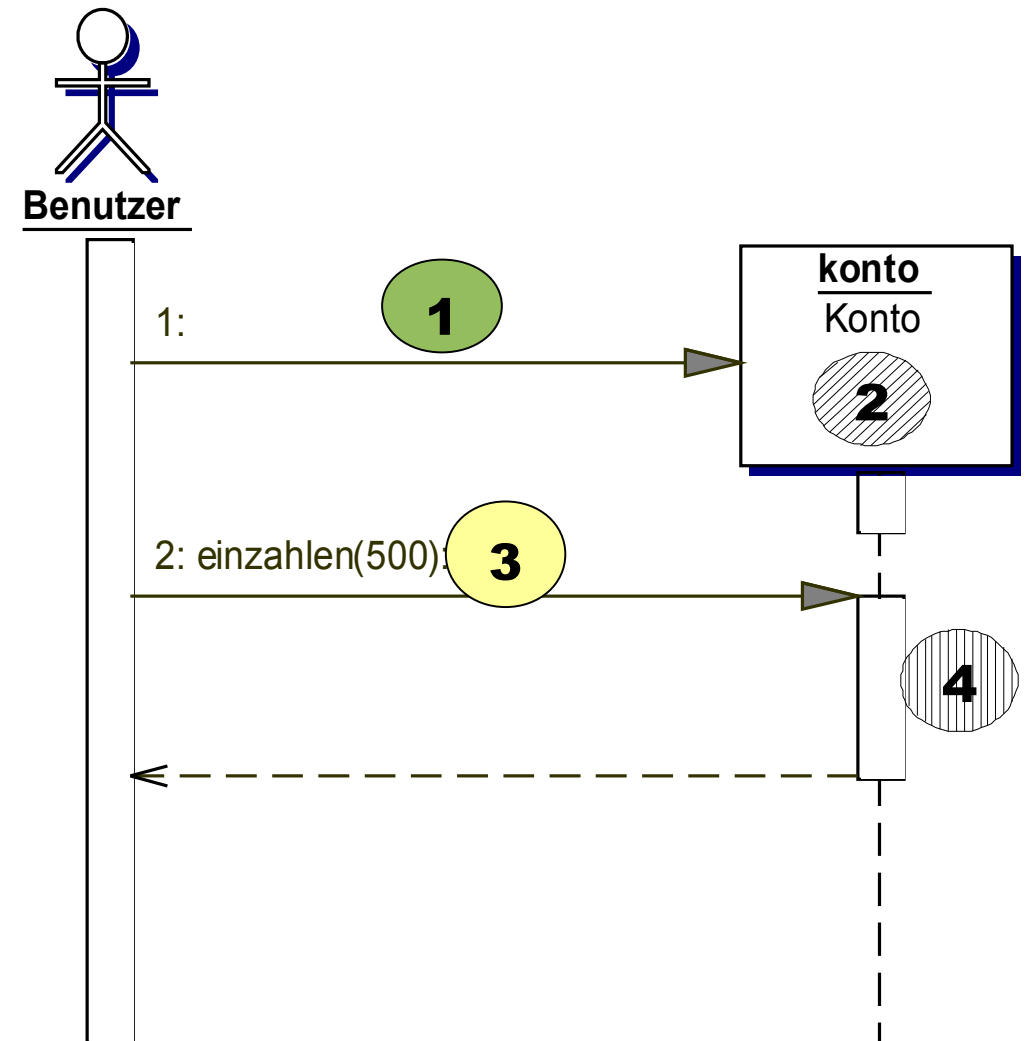
- **Joinpoint**
 - Punkte im Programm, an denen Code erweitert oder modifiziert werden soll
- **Pointcut**
 - eine Auswahl von Joinpoints
- **Advice**
 - der Code für den Pointcut
- **Aspect**
 - Konstrukt, in dem das obige abgelegt wird

Joinpoints können sein

- Aufrufen einer Methode
- Ausführen einer Methode
- Zugriff auf eine Variable
- Behandeln einer Exception
- Initialisierung einer Klasse
- Initialisierung eines Objekts

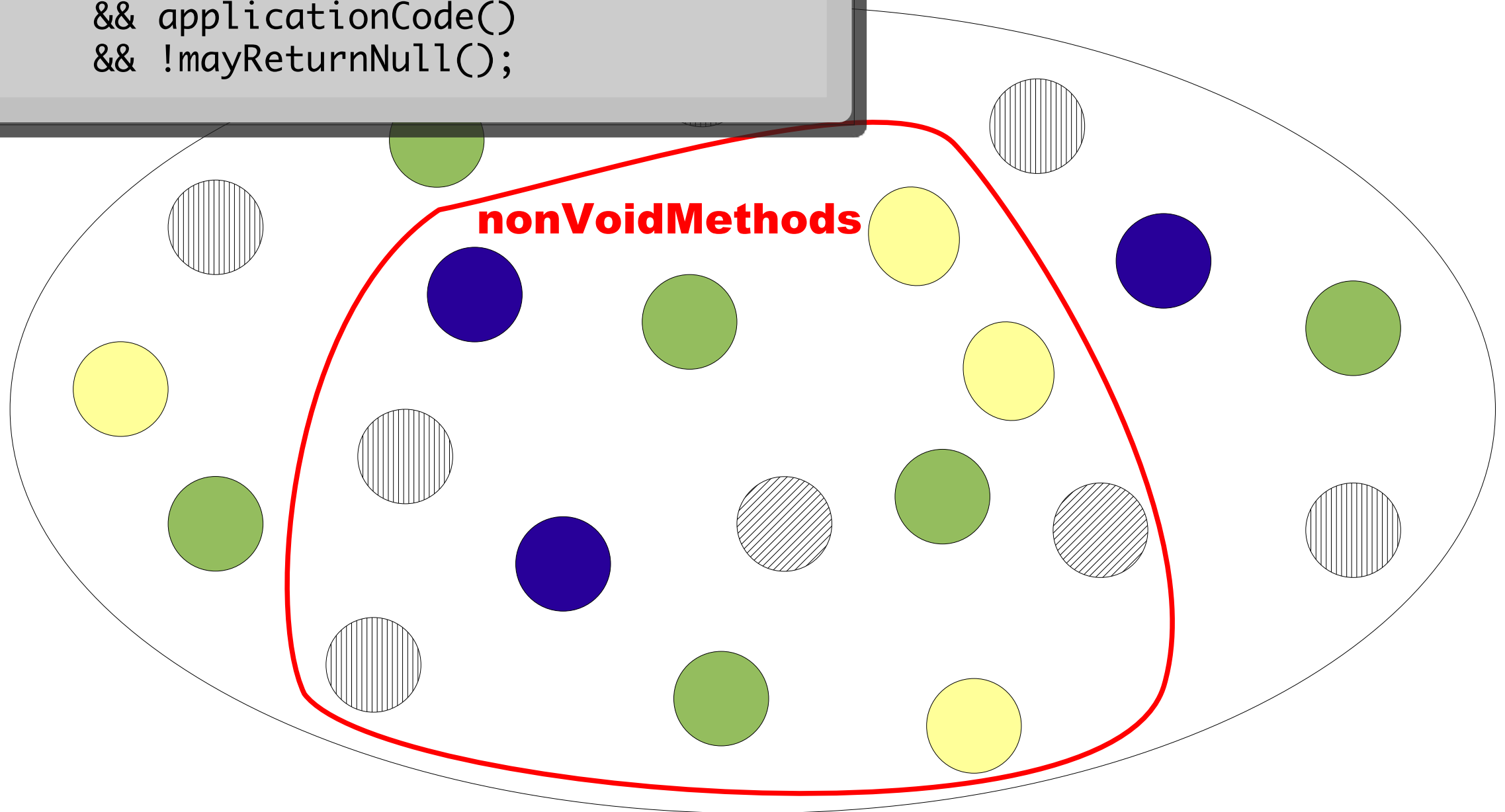
```
Konto konto = new Konto();  
konto.einzahlen(500.0);
```

- (1) Konstruktor aufrufen
- (2) Objekt initialisieren
- (3) Methode aufrufen
- (4) Methode ausführen



Pointcuts

```
private pointcut nonVoidMethods() :  
    execution(public Object+ *.*(..))  
    && applicationCode()  
    && !mayReturnNull();
```



Advice

- Code, der eingewebt wird
 - before()
 - after()
 - around()
- Ähnlichkeit mit Methoden

```
after() returning(Object returned) : nonVoidMethods() {  
    assert returned != null : thisJoinPoint.getSignature()  
        + " returns null!";  
}
```

Your Code

Tools

Check.CT

Check.RT

Exception

Concurrent

Samples

Runtime

A red velvet curtain with a scalloped edge hangs across the top of the frame, framing a white background.

**Dritter Akt:
Kein Anschluss
unter dieser Nummer!**

```
int rc = HttpURLConnection.getResponseCode();
```

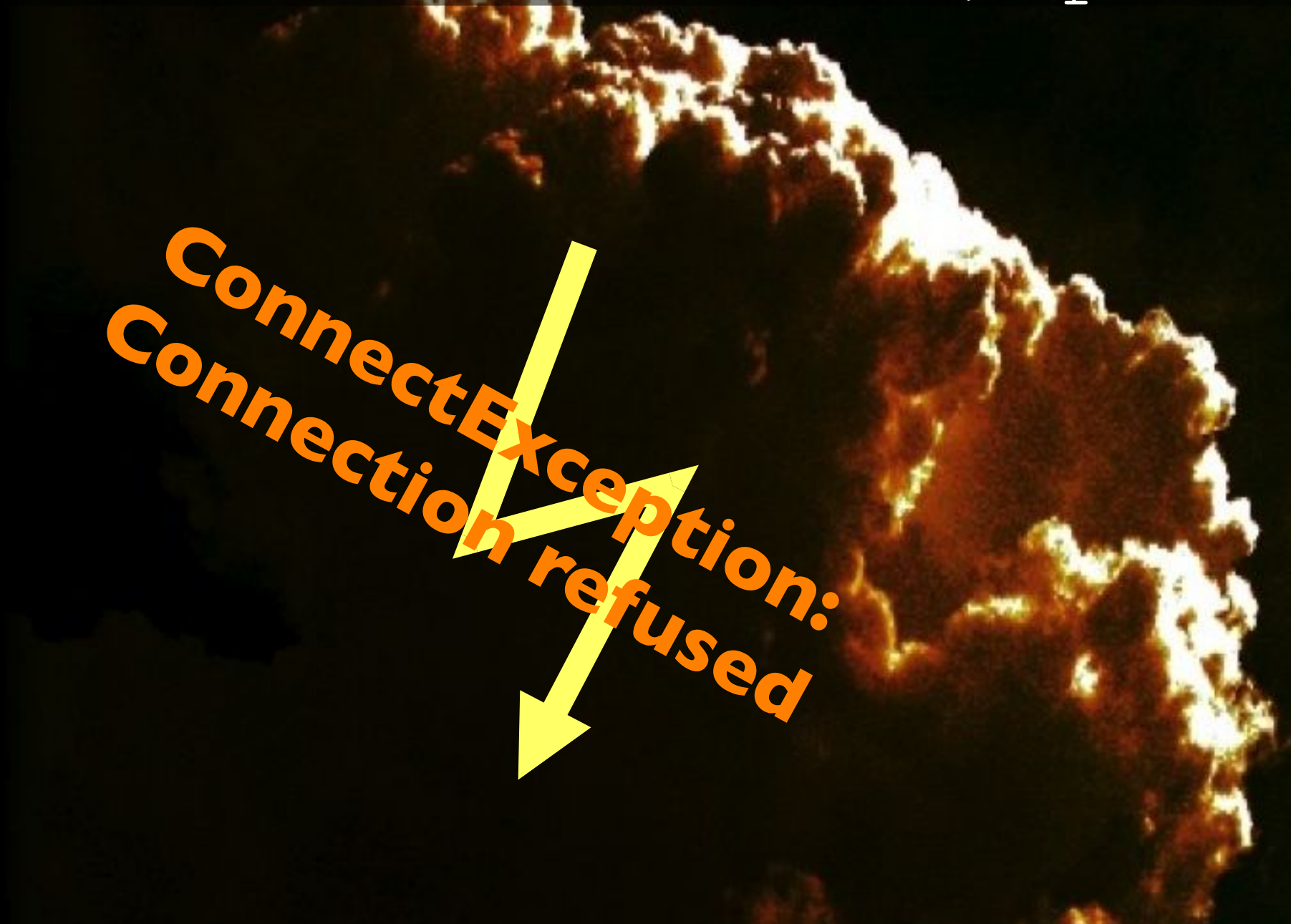
NoRouteToHostException:
No route to host

```
int rc = HttpURLConnection.getResponseCode();
```

NoRouteToHostException:
No route to 216.34.181.96

```
socket = new Socket(host, port);
```

**ConnectException:
Connection refused**



```
socket = new Socket(host, port);
```

ConnectException:
Connection to bnd:815 refused



A red curtain with a scalloped edge hangs across the top of the frame, framing a white background.

Vierter Akt: Testen in vollen Zügen

```
@RunWith(SmokeRunner.class)
public final class LoginServletTest {

    ...

    @IntegrationTest("needs LoginService")
    @Test
    public void testDoPost() throws IOException {

        ...
    }
}
```

Run Configurations

Create, manage, and run configurations

Create a configuration that will launch a JUnit test.



type filter text

JUnit4RunnerTest
LocalhostTest
LoginServletTest
ParallelRunnerTest
ParallelRunnerTest-integrationT
patterntesting-check-ct
patterntesting-check-rt
patterntesting-check-rt-SmokeT
patterntesting-concurrent
patterntesting-concurrent-integ
patterntesting-exception
patterntesting-rt
patterntesting-rt-it
patterntesting-rt-SmokeTest
patterntesting-samples-intro
patterntesting-samples-jfs2010
patterntesting-samples-jfs2010
patterntesting-tools
patterntesting-tools-integration

Filter matched 122 of 131 items

Name: patterntesting-samples-jfs2010-integrationTest

Test (x)= Arguments Classpath JRE Source »2

Program arguments:

Variables...

VM arguments:

-ea -Dpatterntesting.integrationTest

Variables...

Working directory:

☒ Default: \${workspace_loc:patterntesting-samples-jfs2010}

☐ Other:

Workspace...

File System...

Variables...

Apply

Revert

Close

Run

@SmokeTest





@Broken

@RunTestOn("Mac OS X")

@SkipTestOn("Linux")



@Deprecated



Fünfter Akt: Die Ausnahme von der Ausnahme

Internal Error



An internal error occurred while showing an internal error.
You are recommended to exit the workbench.
Subsequent errors may happen and may terminate the workbench without
warning.
See the .log file for more details.

Do you want to exit the workbench?

Yes

No

@TestException

Exception

Factory

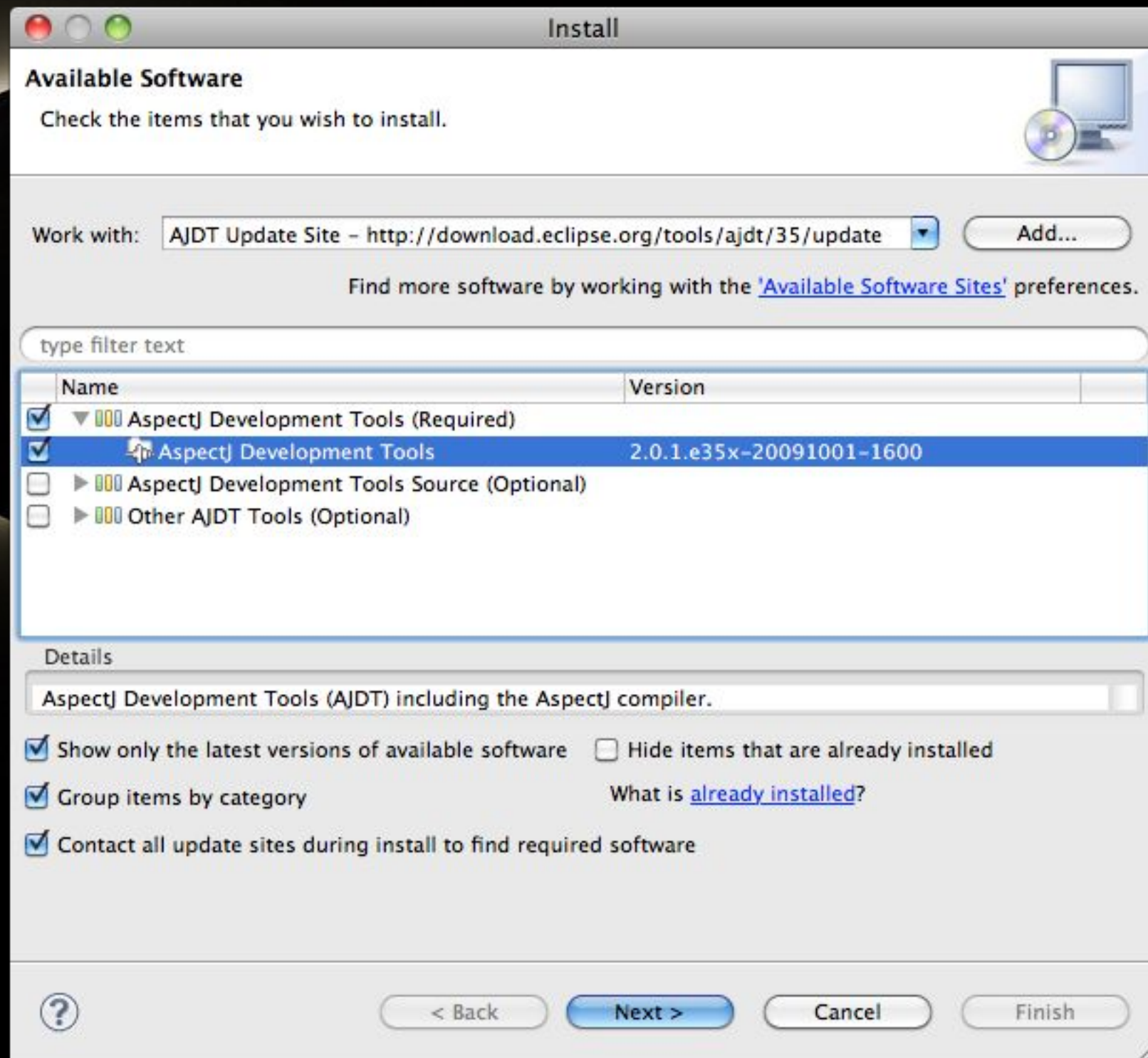
A person with long dark hair is sitting in a meditative pose on a dark, draped surface. The lighting is dramatic, coming from the side, highlighting the contours of their body and hair against a dark background. The person's face is partially obscured by shadow.

**Schlussakt:
mach('s) mit....**

```
<dependency>  
  <groupId>org.patterntesting</groupId>  
  <artifactId>patterntesting-rt</artifactId>  
  <version>1.0.0</version>  
</dependency>
```

1.0.0 noch nicht im Maven Central Repository (30.6.)

```
<repositories>  
  <repository>  
    <id>agentes</id>  
    <name>agentes Repository</name>  
    <url>http://labs.agentes.de/repository</url>  
  </repository>  
</repositories>
```



Einbindung LIGHT



Java

patterntesting-rt.jar

ASPECTJRT_LIB
(aspectjrt.jar)

@RunWith(**SmokeRunner.class**)

@Broken

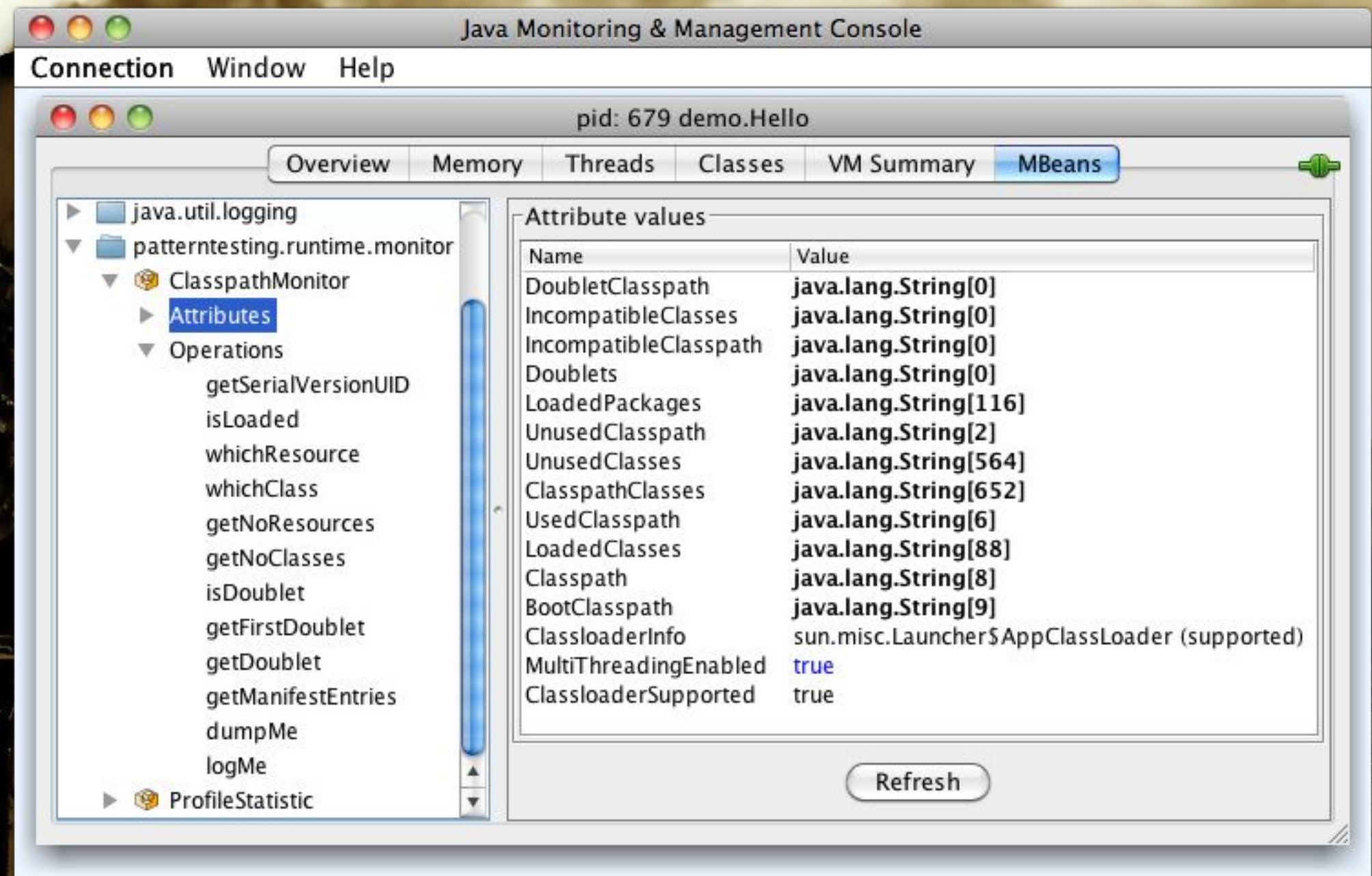
@SkipTestOn

@RunTestOn

@SmokeTest

@IntegrationTest

Pattern Testing *Runtime*



The screenshot shows the Java Monitoring & Management Console window. The title bar reads "Java Monitoring & Management Console". Below the title bar is a menu bar with "Connection", "Window", and "Help". The main window has a sub-title bar "pid: 679 demo.Hello". The "MBeans" tab is selected, showing a tree view on the left and a table of attribute values on the right.

Tree View:

- java.util.logging
- patternesting.runtime.monitor
 - ClasspathMonitor
 - Attributes
 - Operations
 - getSerialVersionUID
 - isLoading
 - whichResource
 - whichClass
 - getNoResources
 - getNoClasses
 - isDoublet
 - getFirstDoublet
 - getDoublet
 - getManifestEntries
 - dumpMe
 - logMe
 - ProfileStatistic

Attribute values table:

Name	Value
DoubletClasspath	java.lang.String[0]
IncompatibleClasses	java.lang.String[0]
IncompatibleClasspath	java.lang.String[0]
Doublets	java.lang.String[0]
LoadedPackages	java.lang.String[116]
UnusedClasspath	java.lang.String[2]
UnusedClasses	java.lang.String[564]
ClasspathClasses	java.lang.String[652]
UsedClasspath	java.lang.String[6]
LoadedClasses	java.lang.String[88]
Classpath	java.lang.String[8]
BootClasspath	java.lang.String[9]
ClassLoaderInfo	sun.misc.Launcher\$AppClassLoader (supported)
MultiThreadingEnabled	true
ClassLoaderSupported	true

Refresh

```
ClasspathMonitor.registerAsMBean();
```

Normal



patterntesting-rt.jar

+ weitere...

Aspect Path



Test-Version

- AspectJ
- patterntesting-rt.jar
- patterntesting-check-rt.jar
- u.a.
- andere Libs

★ Release-Version

- Pure Java
- patterntesting-rt.jar
- andere Libs



@NullAllowed

@MayReturnNull

@LogRuntimeException

@NotYetImplemented

@PublicForTesting

@TestThread

@TestException

@ProfileMe

A red scalloped valance hangs at the top of the frame, with white curtains visible on the left and right sides. The background is a plain white surface.

Zugabe?



@RunWith(ParallelRunner.class)



1.0.0

1.0.1



- ▶ Dokumentation
- ▶ Wiki
- ▶ PatternTesting Samples
- ▶ patterntesting-all
- ▶ Multi-Threading

WANTED

**Java-Entwickler
AspectJ-Entwickler
Maven-Experte**

Sachdienliche Hinweise bitte an:
jobs@patterntesting.de

The Vision



A large blue ship is docked at a wooden pier. The ship has a black anchor on its side and a white conical buoy in the foreground. A yellow 3D text overlay reads 'Tragen?'. The background shows a harbor with other ships and a blue sky.

To be the Minesweeper
for Your Java Code!

Tragen?

in den Hauptrollen:

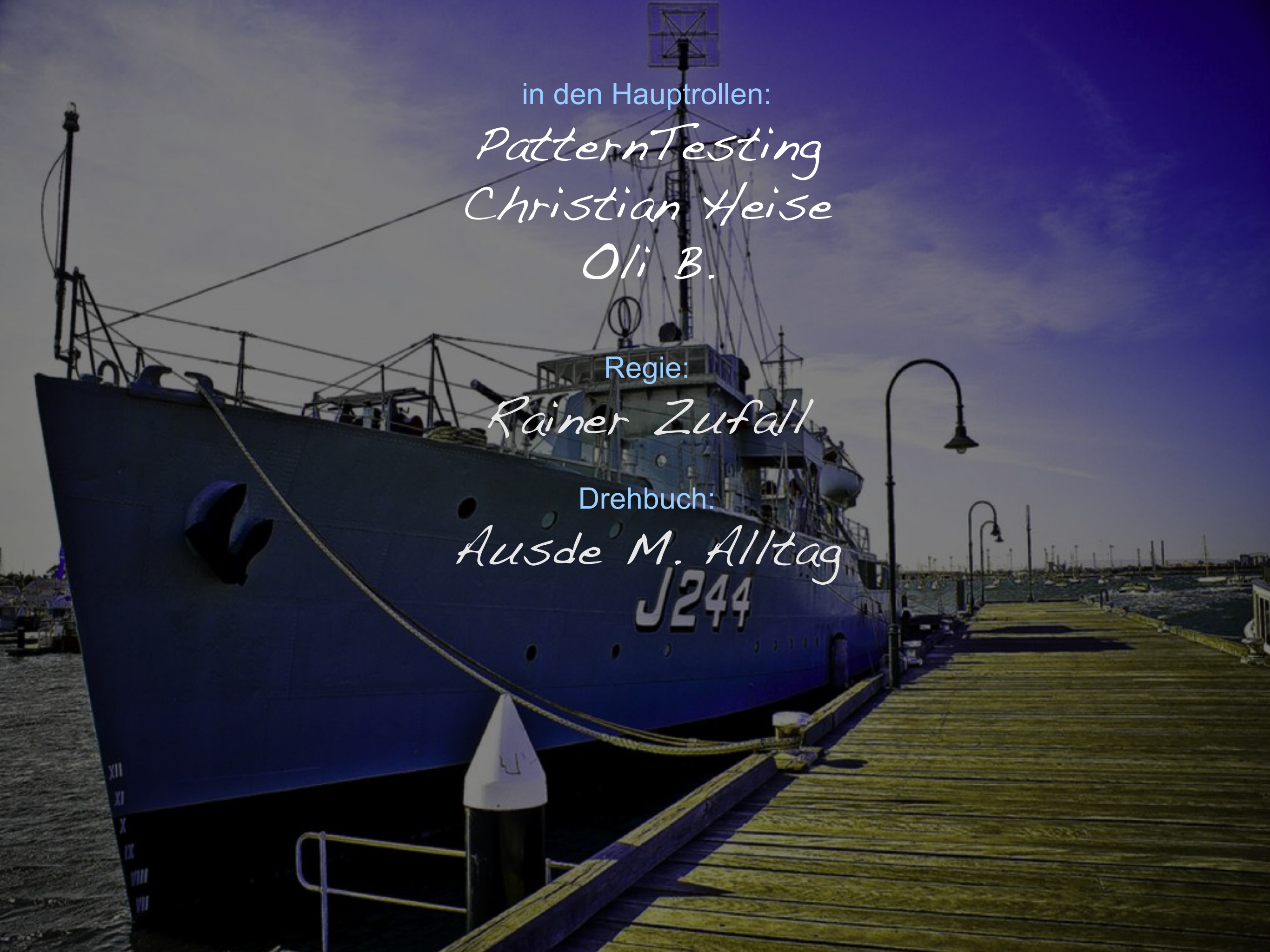
PatternTesting
Christian Heise
Oli B.

Regie:

Rainer Zufall

Drehbuch:

Ausde M. Alltag
J244



Pleiten, Pech und PatternTesting
wurde ihnen präsentiert von

Chris & Oli



weitere Infos:



oliver.boehm@agentes.de

christian.heise@agentes.de

<http://oli.blogger.de>

<http://patternesting.org>